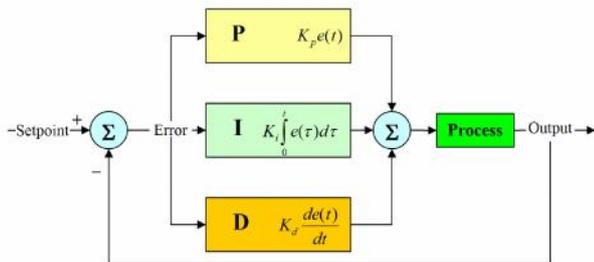




B. PID Controller

PID adalah singkatan dari Proporsional Integral dan Derivatif. PID merupakan suatu sistem kontrol yang digunakan untuk menentukan kontrol presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik (*Feedback*) pada sistem tersebut.



Gambar 2. Blok Diagram PID  
(Sumber: Radhesh.2008)

Ketika membuat suatu sistem kontrol, tentunya ada suatu hal yang menjadi tujuan. Tujuan dari pengontrolan suatu sistem tersebut biasa disebut dengan *setpoint*. sistem akan terus berusaha agar nilai dari setpoint tersebut selalu terpenuhi. Simpangan nilai antara setpoint dengan nilai kenyataan disebut dengan *error*. Nilai error disini didapatkan melalui rumus:

$$Error = Setpoint - Process Variable \quad (1)$$

Sifat-sifat dan karakteristik dari masing-masing kontroler adalah: [3].

a) Proporsional

Pengaruh kontroler proporsional pada sistem adalah

- Dapat menambah atau mengurangi kestabilan sistem
- Dapat memperbaiki respon transien khususnya Rise time dan Setling time
- Dapat mengurangi (bukan menghilangkan) Error steady state.
- Untuk menghilangkan error steady state dibutuhkan nilai konstanta (KP) yang besar yang akan membuat system lebih tidak stabil.
- Kontroler proporsional memberikan pengaruh sebanding dengan nilai error
- Semakin besar error, maka semakin besar sinyal kendali yang di hasilkan oleh kontroler

b) Integral

Pengaruh kontroler integral pada sistem adalah

- Dapat digunakan untuk menghilangkan error steady state
- Respon sistem lebih lambat jika dibandingkan dengan kontroler proporsional
- Dapat menimbulkan ketidakstabilan karena kontroler ini bersifat menambah orde sistem

- Kontroler integral memberikan respons yang sebanding dengan perubahan error
- Semakin besar nilai error, maka semakin cepat sinyal kontrol berubah

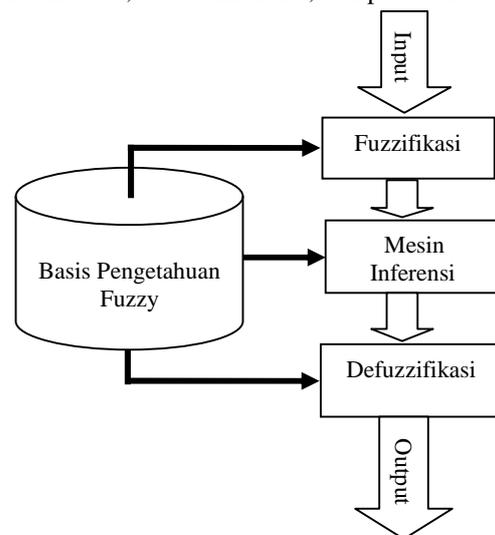
c) Derivatif

Pengaruh kontroler derivatif pada sistem adalah

- Memberikan efek redaman pada sistem yang berosilasi, sehingga dapat memperbesar pemberian nilai KP
- Memperbaiki respon transien, karena memberikan aksi saat ada perubahan error
- Kontroler derivatif hanya akan berubah saat ada perubahan error, sehingga saat ada error statis, kontroler derivatif tidak akan beraksi. Sehingga kontroler derivatif tidak bisa digunakan sendiri, harus digunakan bersamaan dengan kontroler proporsional
- Besarnya sinyal kontrol yang dihasilkan sebanding dengan perubahan error
- Semakin cepat error berubah, maka semakin besar aksi kontrol yang ditimbulkan

C. Fuzzy Logic Control

Berbeda dengan sistem kontrol konvensional yang mendefinisikan sesuatu dengan kategori benar dan salah (1 atau 0) selain nilai itu akan dibulatkan atau diabaikan, logika fuzzy disini dapat mendefinisikan nilai dengan beberapa kategori. Seperti misalkan dalam bahasa keseharian manusia yang menyatakan sesuatu dengan kata “sangat banyak”, “banyak”, “sangat sedikit”, “sangat panas”, “sangat dingin” dan seterusnya. Maka nanti output dari logika fuzzy tidak hanya berupa nilai 1 atau 0, melainkan ada beberapa nilai yang menjadi output yang akan digunakan untuk melakukan proses pada sistem kontrol. Secara umum, struktur dasar logika fuzzy terdiri dari Pengetahuan dasar (*Knowledge Base*), Proses Fuzzifikasi, Mesin Inferensi, dan proses Defuzzifikasi.



Gambar 3. Struktur Dasar Logika Fuzzy

Penjelasan mengenai diagram struktur dasar logika fuzzy adalah sebagai berikut:

1. Basis Pengetahuan Fuzzy

Semua prosedur-prosedur yang akan digunakan dalam proses Fuzzifikasi, Interferensi, dan Defuzzifikasi mengacu pada apa yang ada pada basis pengetahuan fuzzy. Maka dari itu, basis pengetahuan fuzzy sangat berperan penting dalam sistem fuzzy.

2. Fuzzifikasi

Fuzzifikasi adalah bagian dari sistem Fuzzy yang berfungsi untuk merubah nilai yang bersifat pasti (*Crispt*) kedalam variabel linguistik. Proses ini terdiri dari pembentukan membership function yang sesuai dengan basis pengetahuan fuzzy.

3. Proses Inferensi

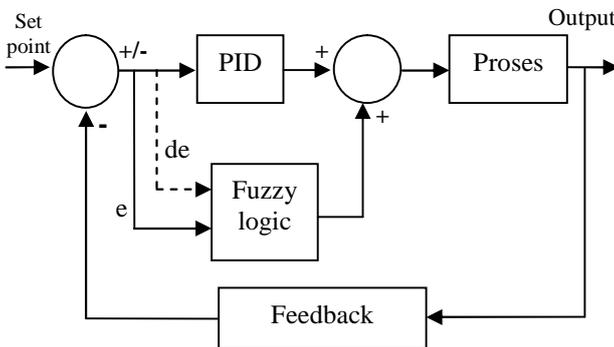
Proses inferensi bertujuan untuk memetakan logika untuk mengambil keputusan. Pada bagian ini biasanya digambarkan dengan table FAM (Fuzzy Associative Map). Pengambilan keputusan untuk output fuzzy disini ditentukan sendiri oleh operator yang membuat logika fuzzy tersebut, dengan mengacu pada basis pengetahuan fuzzy. Biasa proses ini berupa pernyataan implikasi seperti kata Jika-Maka atau yang biasa disebut dengan *IF-Then Rules*[2].

4. Defuzzifikasi

Proses terakhir dalam sistem fuzzy adalah defuzzifikasi, dimana proses ini bertujuan untuk merubah kembali nilai linguistik output dari proses inferensi menjadi nilai pasti (*crispt*) yang akan digunakan pada proses sistem kontrol selanjutnya.

III. PERANCANGAN SISTEM

Sesuai dengan judul diatas, bahwa sistem kontrol pergerakan robot line follower yang akan dibahas pada paper ini berbasis hybrid antara metode kontrol PID dengan metode kontrol Fuzzy logic. Dimana nantinya akan ada dua metode kontrol yang bekerja secara bersama-sama di dalam system.



Gambar 4. Blok Diagram Hybrid PID-Fuzzy Logic

seperti pada dasar teori sistem kontrol, bahwa ada sesuatu yang menjadi tujuan, mengapa sistem tersebut perlu untuk dikontrol. Tujuan tersebut berupa setpoint. Setpoint pada robot line follower adalah suatu keadaan dimana robotdiharuskan untuk tetap berada ditengah-tengah garis.

Misalkan robot line follower memiliki tujuh sensor, maka sensor yang tepat berada ditengah akan menjadi setpoint. Seperti pada gambar 1. diatas, terdapat kondisi sensor dari robot line follower yang tepat berada diatas garis, kondisi tersebut diberikan nilai position value = 0.

A. Perancangan Sistem Kontrol PID

Walaupun sistem kontrol yang digunakan pada pengontrolan pergerakan robot line follower berbasis hybrid, namun tentunya kontroler PID disini masih tetap menjadi kontroler utama [1].

Dalam membuat suatu sistem kontrol yang ditanamkan pada robot tentunya harus diimplementasikan kedalam suatu sintak program. Suatu rumus nantinya akan sedikit berubah penampilan jika telah diimplementasikan ke dalam program, karena pada pemrograman, hanya diperlukan sifat-sifat dan algoritma dari rumus tersebut. Pada perancangan sistem PID ini, akan di jabarkan mengenai pengimplementasian rumus menjadi suatu sintak program sebagai berikut [1]:

a) Proposional

Rumus dasar dari kontroler proposional adalah :

$$P = K_p e(t) \tag{2}$$

Dengan:

- P adalah Proposional
- Kp adalah nilai konstanta proposional
- e(t) adalah nilai error yang selalu akan berubah

Seperti yang dijelaskan pada dasar teori PID, bahwa nilai error didapat dari:

$$e(t) = Sp - Pv \tag{3}$$

Dengan;

- Sp adalah Setpoint
- Pv adalah Process Variable

Jika dirubah kedalam sintak program maka akan menjadi:

```
Setpoint=0
error=Sp-Pv
P=Kp*error
```

b) Integral

Rumus dasar dari kontroler Integral adalah:

$$I = \frac{1}{K_i} \int_0^t e(t) dt \tag{4}$$

Dengan:

- I adalah Integral
- Ki adalah nilai konstanta integral
- e(t) adalah nilai error

dt adalah nilai perubahan waktu (*second*)

Kontroler integral tidak dapat digunakan sendiri, maka dari itu harus digabungkan dengan kontroler proposional.

Jika dirubah kedalam sintak program maka akan menjadi:

```
Setpoint=0
error=Sp-Pv
P=Kp*error
I1=error+last_error
I2=I1*Ki
I=I/5
last_error=error
PI= P+I
```

c) Derivativ

Rumus dasar dari kontroler Derivativ adalah:

$$D = Kd \frac{de(t)}{dt} \tag{5}$$

Dengan:

- D adalah Derivativ
- Kd adalah nilai konstanta Derivativ
- de(t) adalah nilai perubahan error
- dt adalah nilai perubahan waktu (*second*)

Sama juga halnya dengan kontroler Integral, kontroler Derivativ juga tidak bisa digunakan sendirian, melainkan juga harus digabungkan dengan kontroler Proposional. Apabila tidak, maka kontroler Derivativ tidak akan dapat melakukan aksi apapun, karena tidak ada perubahan error.

Jika dirubah kedalam sintak program maka akan menjadi:

```
Setpoint=0
error=Sp-Pv
P=Kp*error
D1=error-last_error
D=D1*Kd
last_error=error
PD=P+D
```

Jadi jika ketiga kontroler tersebut digabungkan, maka sintak programnya akan menjadi:

```
Setpoint=0
error=Sp-Pv
P=Kp*error
I1=error+last_error
I2=I1*Ki
```

```
I=I/5
D1=error-last_error
D=D1*Kd
last_error=error
PD=P+D
PID=PD+I
```

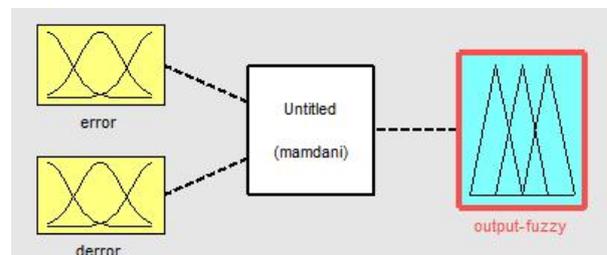
Jadi nilai hasil penjumlahan ketiga kontroler tersebut akan dijumlahkan lagi dengan hasil perhitungan fuzzy logic, yang kemudian akan digunakan untuk mengontrol pergerakan robot.

B. Perancangan Sistem Kontrol Fuzzy Logic

Dalam perancangan sistem okontrol fuzzy logic, tentunya harus mengetahui terlebih dahulu basis pengetahuan fuzzy, mulai dari range semesta pembicaraan (*Universe of Discourse*) yang akan digunakan, range nilai yang akan digunakan pada masing-masing anggota himpunan fuzzy, dan bentuk pola kurva yang akan digunakan.

Metode pemecahan permasalahan fuzzy pada kasus ini adalah menggunakan metode Tsukamoto, karena rumus-rumus yang digunakan mulai dari proses fuzzifikasi sampai proses defuzzifikasi sangat sederhana dan mudah untuk diterapkan pada robot.

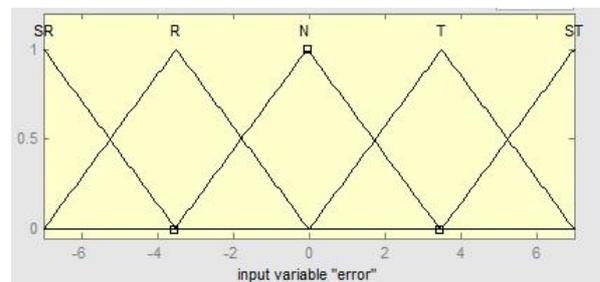
sistem kontrol fuzzy logic pada robot line follower ini menggunakan dua variable input, yaitu error dan derror dan dengan satu variabel output.



Gambar 5. Pemodelan Fuzzy pada Matlab

a) Proses Fuzzifikasi

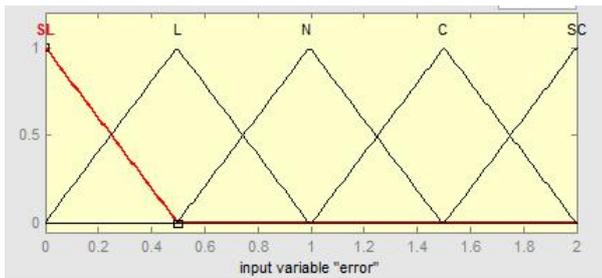
Rancangan membership function dari variable input error adalah:



Gambar 6. Membership Function Variable Error

Keterangan:  
 SR = Sangat Rendah  
 R = Rendah  
 N = Normal  
 T = Tinggi  
 ST = Sangat Tinggi

Rancangan membership function dari variable input derror adalah:



Gambar 7. Membership Function Variable Derror

Keterangan:  
 SL = Sangat Lambat  
 L = Lambat  
 N = Normal  
 C = Cepat  
 SC = Sangat Cepat

Persamaan yang digunakan dalam mencari nilai derajat keanggotaan pada kurva segitiga adalah [2]:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x-a)/(b-a); & a < x < b \\ (b-x)/(c-b); & b < x < c \end{cases} \quad (6)$$

b) Proses Inferensi Pada Tabel FAM

Tabel I. Rule Pada TabelFAM

u_de \ u_e	SR	R	N	T	ST
SL	SS	SS	N	B	B
L	SS	S	N	N	B
N	N	N	N	N	B
C	SS	S	N	B	SB
SC	S	S	B	SB	SB

Keterangan:  
 SS = Sangat Sedikit

S = Sedikit  
 N = Normal  
 B = Banyak  
 SB = Sangat Banyak

Setelah didapatkan aturan-aturan dalam proses inferensi, maka proses selanjutnya adalah merubah nilai-nilai linguistik tersebut kembali ke nilai pasti (Crist Output) melalui proses defuzzifikasi metode rata-rata[3].

Persamaan defuzzifikasi metode Tsukamoto adalah: [2]

$$Z = \frac{\sum_{i=1}^n a_i z_i}{\sum_{i=1}^n a_i} \quad (7)$$

Keterangan:  
 Z = Output Crisp  
 a1 = Member ship function pertama  
 z1 = Nilai hasil inferensi pada member ship function pertama

Jadi setelah kedua sistem dibuat, maka hasil dari kontroler PID dijumlahkan dengan Crisp output dari Fuzzy logic. Sehingga hasil penjumlahan tersebut yang akan digunakan untuk mengendalikan kecepatan motor pada robot Line follower.

Dapat diperlihatkan potongan sintak program penjumlahan dan pengendalian kecepatan putaran motor sebagai berikut:

```

hybrid=PID+Z
motor_kanan=speed+hybrid
motor_kiri=speed-hybrid
    
```

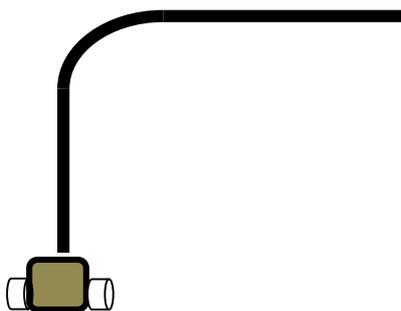
#### IV. HASIL DAN PEMBAHASAN

Berikut ini merupakan hasil pengujian yang dilakukan untuk membandingkan respon antara penggunaan kontroler hybrid dengan penggunaan kontroler PID murni. Analisa ini dilakukan dengan cara menguji pergerakan robot pada track yang memiliki satu buah tikungan. Tikungan tersebut diasumsikan adalah sebuah gangguan. Adapun parameter yang dijadikan perbandingan adalah nilai waktu untuk mencapai keadaan steady state dari respon sistem. Mula-mula robot bergerak lurus dengan kecepatan tertentu, kemudian secara tiba-tiba robot diberikan tikungan. Pada saat robot mengetahui adanya gangguan yang berupa tikungan, maka sistem akan mulai merekam nilai waktu yang dibutuhkan oleh robot untuk kembali stabil. Adapun robot line follower yang digunakan dalam pengujian adalah:



Gambar 8. Robot Yang Digunakan Pengujian

Adapun gambaran bentuk track yang digunakan untuk pengujian adalah sebagai berikut:



Gambar 9. Bentuk Track Yang digunakan dalam uji coba

Tabel II. Hasil Pengujian Nilai Waktu Respon

No	Input				Output	
	Kecepatan	KP	KD	KI	Hybrid	PID
1	80	20	30	1	-	1190ms
2	100	20	30	2	-	1045ms
3	110	30	40	3	-	905ms
4	120	30	40	3	-	889ms
5	120	40	50	4	-	883ms
6	80	20	30	1	699ms	-
7	100	20	30	2	643ms	-
8	110	30	40	3	598ms	-
9	120	30	40	3	556ms	-
10	120	40	50	4	403ms	-

Keterangan dari tabel tersebut adalah:

- Pengujian Hybrid
- Pengujian PID murni

- \*Nilai parameter kecepatan adalah nilai PWM
- \*Satuan output dari pengujian adalah milli second (ms)

Dari hasil pengujian diatas dapat dibandingkan antara menggunakan kendali PID murni dengan menggunakan kendali hybrid dapat memberikan respon pergerakan robot yang lebih cepat dihitung mulai dari robot pertama kali terkena gangguan sampai robot kembali stabil pada garis.

### V. SIMPULAN

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa, metode fuzzy logic yang digabungkan dengan kontroler PID dapat diterapkan pada kendali pergerakan robot line follower untuk mempercepat nilai recovery time dari sistem. Karena, dari semesta nilai derror dan error yang ada, telah dipetakan sedemikian rupa kedalam derajat keanggotaan fuzzy, sehingga tanggapan respon lebih cepat.

### DAFTAR PUSTAKA

- [1] Fahmizal. 2011. *Robot Line Follower dengan Kendali PID-Fuzzy*. (www.fahmizaleeits.wordpress.com, diakses 8 Oktober 2013).
- [2] Sukmadana, B. 2011. "Rancang Bangun Robot Mobile Menggunakan Logika Fuzzy Untuk Bernavigasi Berbasis Mikrokontroler AVR Atmega 8535". *Dielektrika, ISSN 2086-9487 Vol.2, No. 1:9-17, Februari 2011*
- [3] Sutejo, T. , Mulyanto, E. & Suhartono, V. 2011. *Kecerdasan Buatan*. Yogyakarta: Andi Offset.
- [4] Ogata, K. 2002. *Modern Control Engineering Fourth Edition*. Tehran: Aeeizh
- [5] Prabowo, Y. 2013. "Aplikasi PID pada Robot Line Follower Berbasis Mikrokontroler AT-8535". *Arsitron, Vol. 4 No.1 Juni 2013*